GLAST The Gamma Ray Large Area Space Telescope

GLAST LAT ACD ELECTRONICS SUBSYSTEM TECHNICAL DOCUMENT	Document # **ACD-PROC-000149**	Date Effective 10-17-2003
	Prepared by: J. L. Odom and D. A. Sheppard	Supersedes None
Document Title **GLAST LAT ACD GARC Short Functional Test Procedure**		

Gamma-ray Large Area Space Telescope (GLAST)
Large Area Telescope (LAT)
Anti-Coincidence Detector (ACD)

GARC ASIC (GLAST ACD Readout Controller)

Short Functional Test

Rev (–) October 17, 2003

Document Approval

Prepared by: ______________________________ James Odom Date ACD Electronics	Prepared by: ______________________________ Dave Sheppard Date ACD Electronics
Approved by: ______________________________ Glenn Unger Date ACD Electronics System Lead	Approved by: ______________________________ George Shiblie Date ACD Systems Engineer
Approved by: ______________________________ Tom Johnson Date ACD Project Manager	Approved by: ______________________________ Dave Thompson Date ACD Subsystems Manager
Approved by: ______________________________ Eileen Fowler Date Quality Assurance	Approved by: ______________________________ Bob Hartman Date ACD Scientist

GARC Test Table of Contents

Document Change Log

Revision	Date	Change Description	Prepared by
Initial Draft	September 23, 2003	Updates from first GARC testing	Jim Odom and Dave Sheppard
(-)	October 17, 2003	Final Release	Jim Odom and Dave Sheppard

1.0 Description of the GARC ASIC

GARC is the acronym for the Gamma-Ray Large Area Space Telescope (GLAST) Anti-Coincidence Detector (ACD) Readout Controller Application Specific Integrated Circuit (ASIC). The GARC is the main logical interface for the ACD to the LAT instrument electronics. GARC provides command and data return functions for electronics boards associated with the ACD. There are 12 GARCs in the flight system, one for each of the event electronics cards. The GARC is a +3.3V single supply device and communicates to the LAT digitally via LVDS interfaces. The GARC is packaged in a 208 pin plastic quad flatpack.

This document describes the short functional test for the GARC ASIC. The GARC is designed to meet the requirements of the ACD Level IV Electronics Requirements, LAT-SS-00352, and the ACD-LAT Interface Control Document, LAT-SS-00-363.

The design was done utilizing Verilog as the description language, Exemplar Leonardo Spectrum as the synthesis tool targeting the Tanner Agilent 0.5 μm standard cell library, and Tanner L-Edit as the automated place and route layout tool. Core verification was performed via the Tanner LVS tool.

2.0 Preconditions to and Preparations for Starting this Test

This test should only be used after the GARC Test Procedure ACD-PROC-000062 has shown that the GARC design meets the specified requirements. Prior to starting this test, the power supply to the GARC test board should be verified to be at the correct voltage, which is nominally +3.3V. The ACD clock frequency is to be 20 MHz.

Notify QA 24 hours prior to the start of this test. QA will verify test equipment set up and calibration, review documentation and decide if their presence is required during the testing. Indicate on the test record whether QA was present for the testing.

Additionally, prior to starting the functional test, the proper biasing of the GARC circuitry is to be verified. The following table details the expected biases.

GARC Bias Signal	GARC Pin	Nominal Bias Resistor Required	GARC Test Board Res	Function of Bias Resistor
HLD_WOR_BIAS	104	7.50 kΩ to DGND	R19	HLD Wired-OR Rcvr Bias
BIAS_RCVR	156	82.5 kΩ to VDD	R16	LVDS Rcvr Bias Adjust
BIAS_DRV_H	160	4.53 kΩ to DGND	R15	LVDS Driver Bias Adjust
BIAS_DRV_L	169	15.0 kΩ to DGND	R13	LLDRV Driver Bias Adjust
LVDS_PRESETADJ	184	3.92 kΩ to VDD	R20	LVDS Preset Adjust

An additional variation possible to this test is to individually vary the bias resistors by some nominal amount (e.g., 10%) and utilize this functional test to determine variations in performance, if any.

All measurements and test results will be recorded in the “Test Results Record” Appendix 5. A copy of the entire this entire document is not required each time the test is performed. The appendix will serve as the official record each time this test is performed.

2.1 GARC Chip Identification

The test conductor for this test is: ____________________________

Date of the Test: ____________________________

The identification/Serial Number listed on the GARC ASIC is: ___________________________

The serial number of the AEM Simulator board is: ___________________________

The serial number of the GARC Test board is: ___________________________

The serial number of the GAFE Simulator board is: ___________________________

2.2 Test Equipment Utilized

Prior to starting this test, the Test Conductor shall record the test and measurement hardware used in the performance of this test. Note that all multimeters used in testing of the FREE circuit card assembly shall be of low current output design, such as the Fluke 70 series, HP3400 series or similar. High current output multimeters are not compatible with the FREE circuitry.

Instrument Type	Manufacturer & Model Number	NASA ID Number	Calibration Due Date
Power Supply +3.3V			
Power Supply +5.0V			
Multimeter 1			
Multimeter 2			
Pulse Generator			
Oscilloscope			

3.0 GARC Command Mnemonics and Functions

The following table represents each of the available GARC commands. Additionally, all GAFE commands are passed through the GARC. These command patterns are detailed in the document discussing the GAFE logic. There are two types of GARC commands – trigger commands (4 bits in length) and configuration commands (34 bits in length). The AEM-ACD ICD contains the authoritive formats for all command types, but they are repeated in the appendices of this document for convenience. The table below defines the command mnemonics that will be used in this test.

Note that a GAFE will process a write command either for the address hard-wired to the chip address pins or to an address of 'h1F, the GAFE broadcast address. A GAFE will process a read command only for an address identical to the hard-wired address. It is an operational constraint that, for any given ACD circuit board, each GAFE must have a unique address.

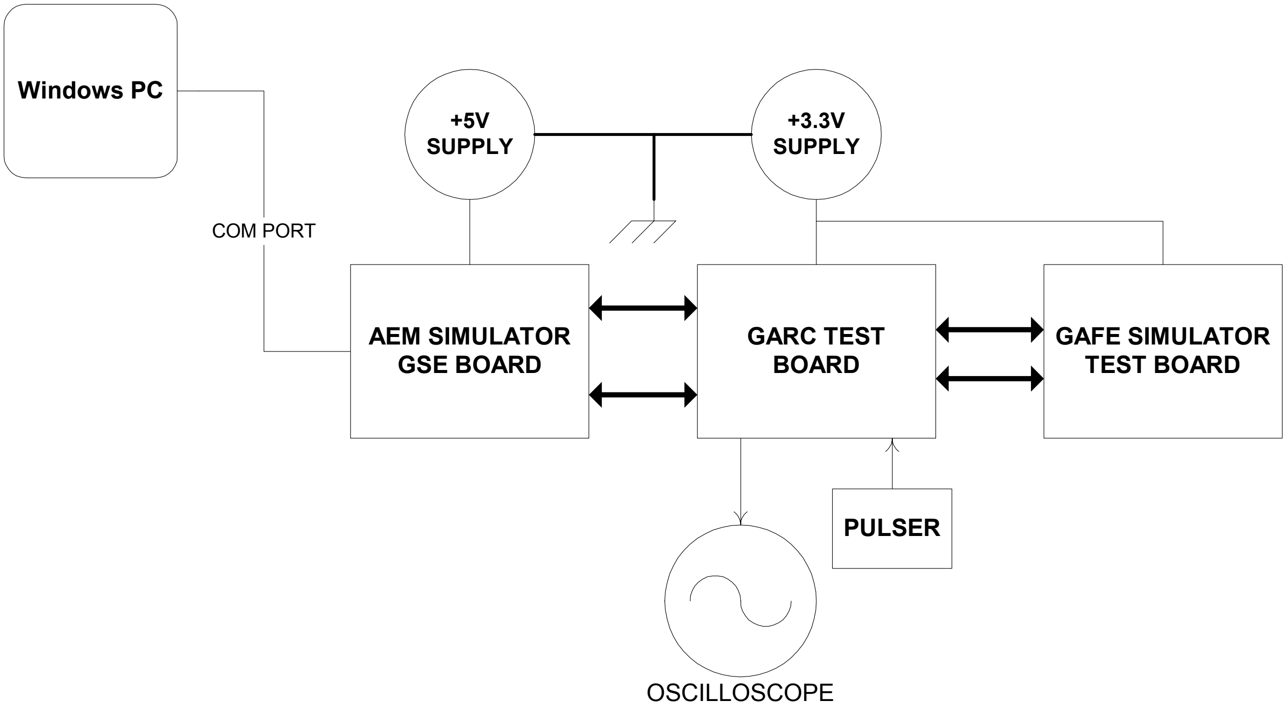
GARC Cmd No.	ACD Command Mnemonic	Rd/Wr Status	Select GARC=0 GAFE=1	Function Block	Register Number	No. of Data Bits	Command Function
1	Trigger_ZS	N/A	N/A	N/A	N/A	N/A	ACD Trigger, Zero-Suppression Enable
2	Trigger_NOZS	N/A	N/A	N/A	N/A	N/A	ACD Trigger, No Zero-Suppression
3	GARC_Reset	W	0	0	1	0	Generates reset for GARC and GAFE registers
4	Veto_Delay_Wr	W	0	0	2	5	Sets Delay from Disc_In to VETO Out
5	Veto_Delay_Rd	R	0	0	2	5	Reads contents of Veto_Delay register
6	GARC_Cal_Strobe	W	0	0	3	0	Sends Calibration Strobe signal to all GAFEs
7	HVBS_Level_Wr	W	0	0	8	12	Sets GARC register value from which HVBS may be commanded in the science mode
8	HVBS_Level_Rd	R	0	0	8	12	Reads contents of HVBS Level register
9	SAA_Level_Wr	W	0	0	9	12	Sets GARC register value from which HVBS may be commanded when in the SAA
10	SAA_Level_Rd	R	0	0	9	12	Reads contents of SAA Level register
11	Use_HV_Normal	W	0	0	10	0	Sends 12 bit value in HVBS Level register to the MAX5121 DAC
12	DAC_HVReg_Rd	R	0	0	10	0	Reads MAX5121 DAC Config Register
13	Use_SAA_Normal	W	0	0	11	0	Sends 12 bit value in SAA Level register to the MAX5121 DAC
14	DAC_SAAReg_Rd	R	0	0	11	0	Reads MAX5121 DAC Config Register
15	Hold_Delay_Wr	W	0	0	12	7	Sets GARC Hold Delay
16	Hold_Delay_Rd	R	0	0	12	7	Reads back value of GARC Hold Delay register
17	Veto_Width_Wr	W	0	0	13	3	Sets width of the VETO signals
18	Veto_Width_Rd	R	0	0	13	3	Reads back width of the VETO width register
19	HitMap_Width_Wr	W	0	0	14	4	Sets width of HitMap pulses
20	HitMap_Width_Rd	R	0	0	14	4	Reads back value in the HitMap width register
21	HitMap_Deadtime_Wr	W	0	0	15	3	Sets stretch at end of HitMap pulses
22	HitMap_Deadtime_Rd	R	0	0	15	3	Sets stretch at end of HitMap pulses
23	Look_At_Me	W	0	1	4	16	GARC interface selection command (primary/secondary)
24	HitMap_Delay_Wr	W	0	1	8	5	Sets delay from Disc_In to HitMap pulse
25	HitMap_Delay_Rd	R	0	1	8	5	Reads back contents of the HitMap_Delay register
26	PHA_EN0_Wr	W	0	1	9	16	Enables and Disables PHA readout enable, channels 0 -15

27	PHA_EN0_Rd	R	0	1	9	16	Reads back contents of the PHA_EN0 register
28	VETO_EN0_Wr	W	0	1	10	16	Enables and Disables VETO signals, channels 0 -15
29	VETO_EN0_Rd	R	0	1	10	16	Reads back contents of the VETO_EN0 register
30	PHA_EN1_Wr	W	0	1	12	2	Enables and Disables PHA readout enable, channels 16 & 17
31	PHA_EN1_Rd	R	0	1	12	2	Reads back contents of the PHA_EN1 register
32	VETO_EN1_Wr	W	0	1	13	2	Enables and Disables VETO signals, channels 16 & 17
33	VETO_EN1_Rd	R	0	1	13	2	Reads back contents of the VETP_EN1 register
34	Max_PHA_Wr	W	0	1	15	5	Sets the Maximum number (limit) of PHA to be returned in a single event data packet
35	Max_PHA_Rd	R	0	1	15	5	Reads back the contents of the Max_PHA register
36	GARC_Mode_Wr	W	0	2	8	11	Sets values of GARC mode bits
37	GARC_Mode_Rd	R	0	2	8	11	Reads back the value of the GARC mode register
38	GARC_Status	R	0	2	9	6	Reads back the value of the GARC status register
39	GARC_Cmd_Reg	R	0	2	10	16	Reads back the value of the GARC command register
40	GARC_Diagnostic	R	0	2	11	16	Reads back the value of the GARC diagnostic register
41	GARC_Cmd_Rejects	R	0	2	12	8	Reads back the number of rejected commands since reset
42	FREE_Board_ID	R	0	2	13	8	Reads back the FREE board serial number
43	GARC_Version	R	0	2	14	3	Reads back the GARC ASIC version number
44	PHA_Thresh00_Wr	W	0	3	8	12	Writes PHA ZS threshold for channel 00
45	PHA_Thresh00_Rd	R	0	3	8	12	Reads back PHA ZS threshold register for channel 00
46	PHA_Thresh01_Wr	W	0	3	9	12	Writes PHA ZS threshold for channel 01
47	PHA_Thresh01_Rd	R	0	3	9	12	Reads back PHA ZS threshold register for channel 01
48	PHA_Thresh02_Wr	W	0	3	10	12	Writes PHA ZS threshold for channel 02
49	PHA_Thresh02_Rd	R	0	3	10	12	Reads back PHA ZS threshold register for channel 02
50	PHA_Thresh03_Wr	W	0	3	11	12	Writes PHA ZS threshold for channel 03
51	PHA_Thresh03_Rd	R	0	3	11	12	Reads back PHA ZS threshold register for channel 03
52	PHA_Thresh04_Wr	W	0	3	12	12	Writes PHA ZS threshold for channel 04
53	PHA_Thresh04_Rd	R	0	3	12	12	Reads back PHA ZS threshold register for channel 04
54	PHA_Thresh05_Wr	W	0	3	13	12	Writes PHA ZS threshold for channel 05
55	PHA_Thresh05_Rd	R	0	3	13	12	Reads back PHA ZS threshold register for channel 05
56	PHA_Thresh06_Wr	W	0	3	14	12	Writes PHA ZS threshold for channel 06
57	PHA_Thresh06_Rd	R	0	3	14	12	Reads back PHA ZS threshold register for channel 06
58	PHA_Thresh07_Wr	W	0	4	8	12	Writes PHA ZS threshold for channel 07
59	PHA_Thresh07_Rd	R	0	4	8	12	Reads back PHA ZS threshold register for channel 07
60	PHA_Thresh08_Wr	W	0	4	9	12	Writes PHA ZS threshold for channel 08
61	PHA_Thresh08_Rd	R	0	4	9	12	Reads back PHA ZS threshold register for channel 08
62	PHA_Thresh09_Wr	W	0	4	10	12	Writes PHA ZS threshold for channel 09
63	PHA_Thresh09_Rd	R	0	4	10	12	Reads back PHA ZS threshold register

							for channel 09
64	PHA_Thresh10_Wr	W	0	4	11	12	Writes PHA ZS threshold for channel 10
65	PHA_Thresh10_Rd	R	0	4	11	12	Reads back PHA ZS threshold register for channel 10
66	PHA_Thresh11_Wr	W	0	4	12	12	Writes PHA ZS threshold for channel 11
67	PHA_Thresh11_Rd	R	0	4	12	12	Reads back PHA ZS threshold register for channel 11
68	PHA_Thresh12_Wr	W	0	4	13	12	Writes PHA ZS threshold for channel 12
69	PHA_Thresh12_Rd	R	0	4	13	12	Reads back PHA ZS threshold register for channel 12
70	PHA_Thresh13_Wr	W	0	4	14	12	Writes PHA ZS threshold for channel 13
71	PHA_Thresh13_Rd	R	0	4	14	12	Reads back PHA ZS threshold register for channel 13
72	PHA_Thresh14_Wr	W	0	5	8	12	Writes PHA ZS threshold for channel 14
73	PHA_Thresh14_Rd	R	0	5	8	12	Reads back PHA ZS threshold register for channel 14
74	PHA_Thresh15_Wr	W	0	5	9	12	Writes PHA ZS threshold for channel 15
75	PHA_Thresh15_Rd	R	0	5	9	12	Reads back PHA ZS threshold register for channel 15
76	PHA_Thresh16_Wr	W	0	5	10	12	Writes PHA ZS threshold for channel 16
77	PHA_Thresh16_Rd	R	0	5	10	12	Reads back PHA ZS threshold register for channel 16
78	PHA_Thresh17_Wr	W	0	5	11	12	Writes PHA ZS threshold for channel 17
79	PHA_Thresh17_Rd	R	0	5	11	12	Reads back PHA ZS threshold register for channel 17
80	ADC_TACQ_Wr	W	0	5	12	6	Sets ADC Acquisition Time from Hold to Start of Conversion
81	ADC_TACQ_Rd	R	0	5	12	6	Reads back contents of the ADC_TACQ register
82	GAFE_Mode_Wr	W	1	GAFE Addr	0	16	Writes the GAFE mode register for the ASIC addressed
83	GAFE_Mode_Rd	R	1	GAFE Addr	0	16	Reads the GAFE mode register contents for the ASIC addressed
84	GAFE_VETO_Wr	W	1	GAFE Addr	1	6	Writes the DAC1 register in the GAFE addressed
85	GAFE_VETO_Rd	R	1	GAFE Addr	1	6	Reads back the contents of the DAC1 register in the addressed GAFE
86	GAFE_VERNIER_Wr	W	1	GAFE Addr	2	6	Writes the DAC2 register in the GAFE addressed
87	GAFE_VERNIER_Rd	R	1	GAFE Addr	2	6	Reads back the contents of the DAC2 register in the addressed GAFE
88	GAFE_HLD_Wr	W	1	GAFE Addr	3	6	Writes the DAC3 register in the GAFE addressed
89	GAFE_HLD_Rd	R	1	GAFE Addr	3	6	Reads back the contents of the DAC3 register in the addressed GAFE
90	GAFE_BIAS_Wr	W	1	GAFE Addr	4	6	Writes the DAC4 register in the GAFE addressed
91	GAFE_BIAS_Rd	R	1	GAFE Addr	4	6	Reads back the contents of the DAC4 register in the addressed GAFE
92	GAFE_TCI_Wr	W	1	GAFE Addr	5	6	Writes the DAC5 register in the GAFE addressed
93	GAFE_TCI_Rd	R	1	GAFE Addr	5	6	Reads back the contents of the DAC5 register in the addressed GAFE
94	GAFE_Version	R	1	GAFE Addr	6	3	Reads back the GAFE ASIC version
95	GAFE_Write_Ctr	R	1	GAFE Addr	7	8	Reads back the contents of the GAFE write counter register
96	GAFE_Reject_Ctr	R	1	GAFE Addr	8	8	Reads back the contents of the GAFE command reject register
97	GAFE_Cmd_Ctr	R	1	GAFE Addr	9	8	Reads back the contents of the GAFE command counter
98	GAFE_Chip_Addr	R	1	GAFE Addr	10	5	Reads back the hardwired address of a GAFE ASIC

4.0 Testing the GARC Digital Logic

The GARC logic is based on a command-response protocol and requires an AEM or AEM simulator to access the logic functions. For each of the following commands, the proper GARC and/or GAFE response may be tested. Note that all GAFEs will respond to a broadcast write command (e.g., GAFE address of decimal 31), but a read command requires a unique GAFE address. The following test details the proper sequence for a single GARC ASIC. The steps are numbered for easier reference. One possible test setup is detailed in the diagram below. If available the GARC Parametric test board maybe used for this test. However all GAFE functions will not be able to be verified and those portions of the test maybe skipped, (I.E. GAFE Register readbacks).



GARC ASIC FUNCTIONAL TEST SETUP

4.1 Measurement of the GARC Bias Resistor Voltages

Verify that the GARC power supply is set to +3.3V. Using the GLAST GARC Test Board measure the following bias voltages on the resistors indicated.

At turn-on, measure the +3.3V current: _____________ mA

GARC Bias Signal	GARC Pin	Resistor Test Point	Expected Voltage	Measured Voltage
HLD_WOR_BIAS	104	R19	1.64	
BIAS_RCVR	156	R16	1.10	
BIAS_DRV_H	160	R15	1.53	
BIAS_DRV_L	169	R13	1.75	
LVDS_PRESET_ADJ	184	R20	1.52	

4.2 Initial Reset Test

This section will verify that GARC registers have been properly initialized during a reset command. The following test sequence of commands will perform this verification. This test will also verify that the GARC to AEM command link is functional.

1. Verify the external GARC Power On Reset pulse. Use the LabView GSE, send the Look at me and Read All Values command.
2. Send the Veto_Delay_Rd command. The data field in the return data stream should be 5.
3. Send the HVBS_Level_Rd command. The data field in the return data stream should be 0.
4. Send the SAA_Level_Rd command. The data field in the return data stream should be 0.
5. Send the Hold_Delay_Rd command. The data field in the return data stream should be 28.
6. Send the Veto_Width_Rd command. The data field in the return data stream should be 2.
7. Send the HitMap_Width_Rd command. The data field in the return data stream should be 7.
8. Send the HitMap_Deadtime_Rd command. The data field in the return data stream should be 3.
9. Send the HitMap_Delay_Rd command. The data field in the return data stream should be 16.
10. Send the PHA_En0_Rd command. The data field in the return data stream should be 65535.
11. Send the PHA_En1_Rd command. The data field in the return data stream should be 3.
12. Send the Veto_En0_Rd command. The data field in the return data stream should be 65535.
13. Send the Veto_En1_Rd command. The data field in the return data stream should be 3.
14. Send the Max_PHA_Rd command. The data field in the return data stream should be 4.
15. Send the GARC_Mode_Rd command. The data field in the return data stream should be 768.
16. Send the GARC_Status command. The data field in the return data stream should be 24.
17. Send the GARC_Cmd_Reg command. The data field in the return data stream should be 0.
18. Send the GARC_Cmd_Rejects command. The data field in the return data stream should be 0.
19. Send the GARC_Version command. The data field in the return data stream should be 3 for GARC V3.
20. Send the PHA_Thresh00_Rd command. The data field in the return data stream should be 1114.
21. Send the PHA_Thresh01_Rd command. The data field in the return data stream should be 1114.
22. Send the PHA_Thresh02_Rd command. The data field in the return data stream should be 1114.
23. Send the PHA_Thresh03_Rd command. The data field in the return data stream should be 1114.
24. Send the PHA_Thresh04_Rd command. The data field in the return data stream should be 1114.
25. Send the PHA_Thresh05_Rd command. The data field in the return data stream should be 1114.
26. Send the PHA_Thresh06_Rd command. The data field in the return data stream should be 1114.
27. Send the PHA_Thresh07_Rd command. The data field in the return data stream should be 1114.
28. Send the PHA_Thresh08_Rd command. The data field in the return data stream should be 1114.
29. Send the PHA_Thresh09_Rd command. The data field in the return data stream should be 1114.
30. Send the PHA_Thresh10_Rd command. The data field in the return data stream should be 1114.

31. Send the PHA_Thresh11_Rd command. The data field in the return data stream should be 1114.
32. Send the PHA_Thresh12_Rd command. The data field in the return data stream should be 1114.
33. Send the PHA_Thresh13_Rd command. The data field in the return data stream should be 1114.
34. Send the PHA_Thresh14_Rd command. The data field in the return data stream should be 1114.
35. Send the PHA_Thresh15_Rd command. The data field in the return data stream should be 1114.
36. Send the PHA_Thresh16_Rd command. The data field in the return data stream should be 1114.
37. Send the PHA_Thresh17_Rd command. The data field in the return data stream should be 1114.
38. Send the ADC_TACQ_Rd command. The data field in the return data stream should be 0.
39. Send the Trigger_ZS command (this captures the FREE board ID). Verify the FREE board ID is 165 indicating the GARC Test board.
40. Send the GARC_Reset command (Do Reset button). Verify that the FREE ID = 255.
41. Verify the status of the GARC registers as shown in steps 2 – 38 above.

If all steps above have verified correctly, the GARC reset function has been verified. A summary of the initial reset parameters for GARC and GAFE is detailed below.

Register	Initial Value (decimal)	Initial Value (hex)
Veto_Delay	5	5
HVBS_Level	0	0
SAA_Level	0	0
Hold_Delay	28	1C
Veto_Width	2	2
HitMap_Width	7	7
HitMap_Deadtime	3	3
HitMap_Delay	16	10
PHA_EN0	65535	FFFF
PHA_EN1	3	3
VETO_EN0	65535	FFFF
VETO_EN1	3	3
Max_PHA	4	4
GARC_Mode	768	300
GARC_Status	24	18
Command_Register	0	0
GARC_Diagnostic	0	0
Cmd_Reject_Ctr	0	0
FREE_Board_ID	*	*
GARC_Version	1	1
PHA_Threshold_00	1114	45A
PHA_Threshold_01	1114	45A
PHA_Threshold_02	1114	45A
PHA_Threshold_03	1114	45A
PHA_Threshold_04	1114	45A
PHA_Threshold_05	1114	45A
PHA_Threshold_06	1114	45A
PHA_Threshold_07	1114	45A
PHA_Threshold_08	1114	45A
PHA_Threshold_09	1114	45A
PHA_Threshold_10	1114	45A

PHA_Threshold_11	1114	45A
PHA_Threshold_12	1114	45A
PHA_Threshold_13	1114	45A
PHA_Threshold_14	1114	45A
PHA_Threshold_15	1114	45A
PHA_Threshold_16	1114	45A
PHA_Threshold_17	1114	45A
ADC_TACQ	0	0
GAFE_Mode	48	30
GAFE_DAC1	57	39
GAFE_DAC2	38	26
GAFE_DAC3	55	37
GAFE_DAC4	32	20
GAFE_DAC5	0	0
GAFE_Wr_Ctr	0	0
GAFE_Reject_Ctr	0	0

5.0 GARC Power Measurements

5.1 Measurement at the Nominal Power Supply Voltage

Verify that the GARC power supply is set to +3.30V. After initial power up, measure the +3.30V power supply current to the GARC in the following modes. Record these values in the table below.

1. Send the GARC_Mode_Wr command with a data argument of 768. Send the GARC_Mode_Rd command to verify. Both the primary and secondary LVDS VETO drivers should be enabled. Record the GARC current in the table below.
2. Send the GARC_Mode_Wr command with a data argument of 256. Send the GARC_Mode_Rd command to verify. Only the primary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
3. Send the GARC_Mode_Wr command with a data argument of 512. Send the GARC_Mode_Rd command to verify. Only the secondary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
4. Send the GARC_Mode_Wr command with a data argument of 0. Send the GARC_Mode_Rd command to verify. None of the LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

GARC Mode	GARC_Mode_Wr Data Argument	+3.3V Current Measured (mA)	+3.3V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		185 ± 10mA
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		135 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		135 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		80 ± 10mA

5.2 Measurement at the Minimum Power Supply Voltage

Verify that the GARC power supply is set to +3.0V. After initial power up, measure the +3.0V power supply current to the GARC in the following modes. Record these values in the table below.

1. Send the GARC_Mode_Wr command with a data argument of 768. Send the GARC_Mode_Rd command to verify. Both the primary and secondary LVDS VETO drivers should be enabled. Record the GARC current in the table below.
2. Send the GARC_Mode_Wr command with a data argument of 256. Send the GARC_Mode_Rd command to verify. Only the primary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
3. Send the GARC_Mode_Wr command with a data argument of 512. Send the GARC_Mode_Rd command to verify. Only the secondary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
4. Send the GARC_Mode_Wr command with a data argument of 0. Send the GARC_Mode_Rd command to verify. None of the LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

GARC Mode	GARC_Mode_Wr Data Argument	+3.0V Current Measured (mA)	+3.0V Current Expected (mA)
LVDS “A” Drivers Enabled LVDS “B” Drivers Enabled	768		160 ± 10mA
LVDS “A” Drivers Enabled LVDS “B” Drivers Disabled	256		115 ± 10mA
LVDS “A” Drivers Disabled LVDS “B” Drivers Enabled	512		115 ± 10mA
LVDS “A” Drivers Disabled LVDS “B” Drivers Disabled	0		65 ± 10mA

5.3 Measurement at the Maximum Power Supply Voltage

Verify that the GARC power supply is set to +3.6V. After initial power up, measure the +3.6V power supply current to the GARC in the following modes. Record these values in the table below.

1. Send the GARC_Mode_Wr command with a data argument of 768. Send the GARC_Mode_Rd command to verify. Both the primary and secondary LVDS VETO drivers should be enabled. Record the GARC current in the table below.
2. Send the GARC_Mode_Wr command with a data argument of 256. Send the GARC_Mode_Rd command to verify. Only the primary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
3. Send the GARC_Mode_Wr command with a data argument of 512. Send the GARC_Mode_Rd command to verify. Only the secondary LVDS VETO drivers should now be enabled. Record the GARC current in the table below.
4. Send the GARC_Mode_Wr command with a data argument of 0. Send the GARC_Mode_Rd command to verify. None of the LVDS VETO drivers should now be enabled. Record the GARC current in the table below.

GARC Mode	GARC_Mode_Wr Data Argument	+3.6V Current Measured (mA)	+3.6V Current Expected (mA)
LVDS “A” Drivers Enabled LVDS “B” Drivers Enabled	768		215 ± 10mA
LVDS “A” Drivers Enabled LVDS “B” Drivers Disabled	256		155 ± 10mA
LVDS “A” Drivers Disabled LVDS “B” Drivers Enabled	512		155 ± 10mA
LVDS “A” Drivers Disabled LVDS “B” Drivers Disabled	0		95 ± 10mA

5. Reset the GARC power supply to +3.3V.
6. Send the GARC_Reset command to return the GARC to the initial power-on configuration.

6.0 GARC Register Read/Write and PHA ReadoutTests

This section tests the proper functioning of each bit of the commandable registers in the GARC. The intent is to toggle each bit in a variety of patterns to ensure that all bits are addressable and that there is no stuck-at-fault condition. This can be accomplished using the LabView GSE software by doing the Register Test 3 times. If the GAFE Simulator is present enable all of the GAFE Registers. If no GAFE Simulator skip the GAFE Register readback portion of the test.

If performed via the automated procedure, the next section will be 6.40. The following GARC registers will be tested in this section:

GARC Register Name	Register Width (bits)	Register Type
Veto_Delay	5	Read/Write
HVBS_Level	12	Read/Write
SAA_Level	12	Read/Write
Hold_Delay	7	Read/Write
Veto_Width	3	Read/Write
HitMap_Width	4	Read/Write
HitMap_Deadtime	3	Read/Write
HitMap_Delay	5	Read/Write
PHA_En0	16	Read/Write
PHA_En1	2	Read/Write
VETO_En0	16	Read/Write
VETO_En1	2	Read/Write
Max_PHA	5	Read/Write
GARC_Mode	11	Read/Write
GARC_Status	6	Read Only
Command_Register	16	Read Only
GARC_Diagnostic	16	Read Only
Cmd_Reject_Counter	8	Read Only
FREE_Board_ID	8	Read Only
GARC_Version	3	Read Only
PHA_Threshold_00	16	Read/Write
PHA_Threshold_01	16	Read/Write
PHA_Threshold_02	16	Read/Write
PHA_Threshold_03	16	Read/Write
PHA_Threshold_04	16	Read/Write
PHA_Threshold_05	16	Read/Write
PHA_Threshold_06	16	Read/Write
PHA_Threshold_07	16	Read/Write
PHA_Threshold_08	16	Read/Write
PHA_Threshold_09	16	Read/Write
PHA_Threshold_10	16	Read/Write
PHA_Threshold_11	16	Read/Write
PHA_Threshold_12	16	Read/Write
PHA_Threshold_13	16	Read/Write
PHA_Threshold_14	16	Read/Write
PHA_Threshold_15	16	Read/Write
PHA_Threshold_16	16	Read/Write
PHA_Threshold_17	16	Read/Write
ADC_TACQ	6	Read/Write

6.1 Veto Delay Register Test

1. Send the Veto_Delay_Wr command with a data argument of 5'h0 (0). Send the Veto_Delay_Rd command and read back this commanded data argument.
2. Send the Veto_Delay_Wr command with a data argument of 5'h15 (21). Send the Veto_Delay_Rd command and read back this commanded data argument.
3. Send the Veto_Delay_Wr command with a data argument of 5'h0A (10). Send the Veto_Delay_Rd command and read back this commanded data argument.
4. Send the Veto_Delay_Wr command with a data argument of 5'h1F (31). Send the Veto_Delay_Rd command and read back this commanded data argument.
5. Send the Veto_Delay_Wr command with a data argument of 5'h5 (5). Send the Veto_Delay_Rd command and read back this commanded data argument.

6.2 HVBS Level Register Test

1. Send the HVBS_Level_Wr command with a data argument of 12'h0 (0). Send the HVBS_Level_Rd command and read back this commanded data argument.
2. Send the HVBS_Level_Wr command with a data argument of 12'h555 (1365). Send the HVBS_Level_Rd command and read back this commanded data argument.
3. Send the HVBS_Level_Wr command with a data argument of 12'hAAA (2730). Send the HVBS_Level_Rd command and read back this commanded data argument.
4. Send the HVBS_Level_Wr command with a data argument of 12'hFFF (4095). Send the HVBS_Level_Rd command and read back this commanded data argument.
5. Send the HVBS_Level_Wr command with a data argument of 12'h0 (0). Send the HVBS_Level_Rd command and read back this commanded data argument.

6.3 SAA Level Register Test

1. Send the SAA_Level_Wr command with a data argument of 12'h0 (0). Send the SAA_Level_Rd command and read back this commanded data argument.
2. Send the SAA_Level_Wr command with a data argument of 12'h555 (1365). Send the SAA_Level_Rd command and read back this commanded data argument.
3. Send the SAA_Level_Wr command with a data argument of 12'hAAA (2730). Send the SAA_Level_Rd command and read back this commanded data argument.
4. Send the SAA_Level_Wr command with a data argument of 12'hFFF (4095). Send the SAA_Level_Rd command and read back this commanded data argument.
5. Send the SAA_Level_Wr command with a data argument of 12'h0 (0). Send the SAA_Level_Rd command and read back this commanded data argument.

6.4 Hold Delay Register Test

1. Send the Hold_Delay_Wr command with a data argument of 7'h0 (0). Send the Hold_Delay_Rd command and read back this commanded data argument.

2. Send the Hold_Delay_Wr command with a data argument of 7’h55 (85). Send the Hold_Delay_Rd command and read back this commanded data argument.
3. Send the Hold_Delay_Wr command with a data argument of 7’h2A (42). Send the Hold_Delay_Rd command and read back this commanded data argument.
4. Send the Hold_Delay_Wr command with a data argument of 7’h7F (127). Send the Hold_Delay_Rd command and read back this commanded data argument.
5. Send the Hold_Delay_Wr command with a data argument of 7’h1C (28). Send the Hold_Delay_Rd command and read back this commanded data argument.

6.5 Veto Width Register Test

1. Send the Veto_Width_Wr command with a data argument of 3’h0 (0). Send the Veto_Width_Rd command and read back this commanded data argument.
2. Send the Veto_Width_Wr command with a data argument of 3’h5 (5). Send the Veto_Width_Rd command and read back this commanded data argument.
3. Send the Veto_Width_Wr command with a data argument of 3’h7 (7). Send the Veto_Width_Rd command and read back this commanded data argument.
4. Send the Veto_Width_Wr command with a data argument of 3’h2 (2). Send the Veto_Width_Rd command and read back this commanded data argument.

6.6 HitMap Width Register Test

1. Send the HitMap_Width_Wr command with a data argument of 4’h0 (0). Send the HitMap_Width_Rd command and read back this commanded data argument.
2. Send the HitMap_Width_Wr command with a data argument of 4’h5 (5). Send the HitMap_Width_Rd command and read back this commanded data argument.
3. Send the HitMap_Width_Wr command with a data argument of 4’hA (10). Send the HitMap_Width_Rd command and read back this commanded data argument.
4. Send the HitMap_Width_Wr command with a data argument of 4’hF (15). Send the HitMap_Width_Rd command and read back this commanded data argument.
5. Send the HitMap_Width_Wr command with a data argument of 4’h7 (7). Send the HitMap_Width_Rd command and read back this commanded data argument.

6.7 HitMap Deadtime Register Test

1. Send the HitMap_Deadtime_Wr command with a data argument of 3’h0 (0). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
2. Send the HitMap_Deadtime_Wr command with a data argument of 3’h5 (5). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
3. Send the HitMap_Deadtime_Wr command with a data argument of 3’h2 (2). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
4. Send the HitMap_Deadtime_Wr command with a data argument of 3’h7 (7). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.
5. Send the HitMap_Deadtime_Wr command with a data argument of 3’h3 (3). Send the HitMap_Deadtime_Rd command and read back this commanded data argument.

6.8 HitMap Delay Register Test

1. Send the HitMap_Delay_Wr command with a data argument of 5'h0 (0). Send the HitMap_Delay_Rd command and read back this commanded data argument.
2. Send the HitMap_Delay_Wr command with a data argument of 5'h15 (21). Send the HitMap_Delay_Rd command and read back this commanded data argument.
3. Send the HitMap_Delay_Wr command with a data argument of 5'h0A (10). Send the HitMap_Delay_Rd command and read back this commanded data argument.
4. Send the HitMap_Delay_Wr command with a data argument of 5'h1F (31). Send the HitMap_Delay_Rd command and read back this commanded data argument.
5. Send the HitMap_Delay_Wr command with a data argument of 5'h10 (16). Send the HitMap_Delay_Rd command and read back this commanded data argument.

6.9 PHA En0 Register Test

1. Send the PHA_En0_Wr command with a data argument of 16'h0 (0). Send the PHA_En0_Rd command and read back this commanded data argument.
2. Send the PHA_En0_Wr command with a data argument of 16'h5555 (21845). Send the PHA_En0_Rd command and read back this commanded data argument.
3. Send the PHA_En0_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_En0_Rd command and read back this commanded data argument.
4. Send the PHA_En0_Wr command with a data argument of 16'hFFFF (65535). Send the PHA_En0_Rd command and read back this commanded data argument.

6.10 Veto En0 Register Test

1. Send the VETO_En0_Wr command with a data argument of 16'h0 (0). Send the VETO_En0_Rd command and read back this commanded data argument.
2. Send the VETO_En0_Wr command with a data argument of 16'h5555 (21845). Send the VETO_En0_Rd command and read back this commanded data argument.
3. Send the VETO_En0_Wr command with a data argument of 16'hAAAA (43690). Send the VETO_En0_Rd command and read back this commanded data argument.
4. Send the VETO_En0_Wr command with a data argument of 16'hFFFF (65535). Send the VETO_En0_Rd command and read back this commanded data argument.

6.11 PHA En1 Register Test

1. Send the PHA_En1_Wr command with a data argument of 2'h0 (0). Send the PHA_En1_Rd command and read back this commanded data argument.
2. Send the PHA_En1_Wr command with a data argument of 2'h1 (1). Send the PHA_En1_Rd command and read back this commanded data argument.
3. Send the PHA_En1_Wr command with a data argument of 2'h2 (2). Send the PHA_En1_Rd command and read back this commanded data argument.

4. Send the PHA_En1_Wr command with a data argument of 2’h3 (3). Send the PHA_En1_Rd command and read back this commanded data argument.

6.12 Veto En1 Register Test

1. Send the VETO_En1_Wr command with a data argument of 2’h0 (0). Send the VETO_En1_Rd command and read back this commanded data argument.
2. Send the VETO_En1_Wr command with a data argument of 2’h1 (1). Send the VETO_En1_Rd command and read back this commanded data argument.
3. Send the VETO_En1_Wr command with a data argument of 2’h2 (2). Send the VETO_En1_Rd command and read back this commanded data argument.
4. Send the VETO_En1_Wr command with a data argument of 2’h3 (3). Send the VETO_En1_Rd command and read back this commanded data argument.

6.13 MaxPHA Register Test

1. Send the MaxPHA_Wr command with a data argument of 5’h0 (0). Send the MaxPHA_Rd command and read back this commanded data argument.
2. Send the MaxPHA_Wr command with a data argument of 5’h15 (21). Send the MaxPHA_Rd command and read back this commanded data argument.
3. Send the MaxPHA_Wr command with a data argument of 5’h0A (10). Send the MaxPHA_Rd command and read back this commanded data argument.
4. Send the MaxPHA_Wr command with a data argument of 5’h1F (31). Send the MaxPHA_Rd command and read back this commanded data argument.
5. Send the MaxPHA_Wr command with a data argument of 5’h4 (4). Send the MaxPHA_Rd command and read back this commanded data argument.

6.14 GARC Mode Register Test

**** Note this sequence must be followed exactly to preclude placing the GARC into the undesired mode of sending return data with incorrect parity.**

1. Send the GARC_Mode_Wr command with a data argument of 11’h2 (2). Send the GARC_Mode_Rd command and read back this commanded data argument.
2. Send the GARC_Mode_Wr command with a data argument of 11’h4 (4). Send the GARC_Mode_Rd command and read back this commanded data argument.
3. Send the GARC_Mode_Wr command with a data argument of 11’h8 (8). Send the GARC_Mode_Rd command and read back this commanded data argument.
4. Send the GARC_Mode_Wr command with a data argument of 11’h10 (16). Send the GARC_Mode_Rd command and read back this commanded data argument.
5. Send the GARC_Mode_Wr command with a data argument of 11’h20 (32). Send the GARC_Mode_Rd command and read back this commanded data argument.
6. Send the GARC_Mode_Wr command with a data argument of 11’h40 (64). Send the GARC_Mode_Rd command and read back this commanded data argument.

7. Send the GARC_Mode_Wr command with a data argument of 11'h80 (128). Send the GARC_Mode_Rd command and read back this commanded data argument.
8. Send the GARC_Mode_Wr command with a data argument of 11'h100 (256). Send the GARC_Mode_Rd command and read back this commanded data argument.
9. Send the GARC_Mode_Wr command with a data argument of 11'h200 (512). Send the GARC_Mode_Rd command and read back this commanded data argument.
10. Send the GARC_Mode_Wr command with a data argument of 11'h400 (1024). Send the GARC_Mode_Rd command and read back this commanded data argument.
11. Send the GARC_Mode_Wr command with a data argument of 11'h300 (768). Send the GARC_Mode_Rd command and read back this commanded data argument.

6.15 PHA Threshold Channel 00 Register Test

1. Send the PHA_Thresh00_Wr command with a data argument of 12'h0 (0). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh00_Wr command with a data argument of 12'h555 (21845). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh00_Wr command with a data argument of 16'hAAA (43690). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh00_Wr command with a data argument of 16'hFFF (65536). Send the PHA_Thresh00_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh00_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh00_Rd command and read back this commanded data argument.

6.16 PHA Threshold Channel 01 Register Test

1. Send the PHA_Thresh01_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh01_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh01_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh01_Wr command with a data argument of 16'hAAAA (65536). Send the PHA_Thresh01_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh01_Wr command with a data argument of 16'h45A (1114). Send the PHA_Thresh01_Rd command and read back this commanded data argument.

6.17 PHA Threshold Channel 02 Register Test

1. Send the PHA_Thresh02_Wr command with a data argument of 16'h0 (0). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh02_Wr command with a data argument of 16'h5555 (21845). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh02_Wr command with a data argument of 16'hAAAA (43690). Send the PHA_Thresh02_Rd command and read back this commanded data argument.

4. Send the PHA_Thresh02_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh02_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh02_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh02_Rd command and read back this commanded data argument.

6.18 PHA Threshold Channel 03 Register Test

1. Send the PHA_Thresh03_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh03_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh03_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh03_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh03_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh03_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh03_Rd command and read back this commanded data argument.

6.19 PHA Threshold Channel 04 Register Test

1. Send the PHA_Thresh04_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh04_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh04_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh04_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh04_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh04_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh04_Rd command and read back this commanded data argument.

6.20 PHA Threshold Channel 05 Register Test

1. Send the PHA_Thresh05_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh05_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh05_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh05_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh05_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh05_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh05_Rd command and read back this commanded data argument.

6.21 PHA Threshold Channel 06 Register Test

1. Send the PHA_Thresh06_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh06_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh06_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh06_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh06_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh06_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh06_Rd command and read back this commanded data argument.

6.22 PHA Threshold Channel 07 Register Test

1. Send the PHA_Thresh07_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh07_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh07_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh07_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh07_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh07_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh07_Rd command and read back this commanded data argument.

6.23 PHA Threshold Channel 08 Register Test

1. Send the PHA_Thresh08_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh08_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh08_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh08_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh08_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh08_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh08_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh08_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh08_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh08_Rd command and read back this commanded data argument.

6.24 PHA Threshold Channel 09 Register Test

1. Send the PHA_Thresh09_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh09_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh09_Rd command and read back this commanded data argument.

3. Send the PHA_Thresh09_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh09_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh09_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh09_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh09_Rd command and read back this commanded data argument.

6.25 PHA Threshold Channel 10 Register Test

1. Send the PHA_Thresh10_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh10_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh10_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh10_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh10_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh10_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh10_Rd command and read back this commanded data argument.

6.26 PHA Threshold Channel 11 Register Test

1. Send the PHA_Thresh11_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh11_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh11_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh11_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh11_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh11_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh11_Rd command and read back this commanded data argument.

6.27 PHA Threshold Channel 12 Register Test

1. Send the PHA_Thresh12_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh12_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh12_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh12_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh12_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh12_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh12_Rd command and read back this commanded data argument.

6.28 PHA Threshold Channel 13 Register Test

1. Send the PHA_Thresh13_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh13_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh13_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh13_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh13_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh13_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh13_Rd command and read back this commanded data argument.

6.29 PHA Threshold Channel 14 Register Test

1. Send the PHA_Thresh14_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh14_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh14_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh14_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh14_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh14_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh14_Rd command and read back this commanded data argument.

6.30 PHA Threshold Channel 15 Register Test

1. Send the PHA_Thresh15_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh15_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh15_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh15_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh15_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh15_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh15_Rd command and read back this commanded data argument.

6.31 PHA Threshold Channel 16 Register Test

1. Send the PHA_Thresh16_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh16_Rd command and read back this commanded data argument.

2. Send the PHA_Thresh16_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh16_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh16_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh16_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh16_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh16_Rd command and read back this commanded data argument.

6.32 PHA Threshold Channel 17 Register Test

1. Send the PHA_Thresh17_Wr command with a data argument of 16’h0 (0). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
2. Send the PHA_Thresh17_Wr command with a data argument of 16’h5555 (21845). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
3. Send the PHA_Thresh17_Wr command with a data argument of 16’hAAAA (43690). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
4. Send the PHA_Thresh17_Wr command with a data argument of 16’hAAAA (65536). Send the PHA_Thresh17_Rd command and read back this commanded data argument.
5. Send the PHA_Thresh17_Wr command with a data argument of 16’h45A (1114). Send the PHA_Thresh17_Rd command and read back this commanded data argument.

6.33 ADC TACQ Register Test

1. Send the ADC_TACQ_Wr command with a data argument of 6’h0 (0). Send the ADC_TACQ_Rd command and read back this commanded data argument.
2. Send the ADC_TACQ_Wr command with a data argument of 6’h15 (21). Send the ADC_TACQ_Rd command and read back this commanded data argument.
3. Send the ADC_TACQ_Wr command with a data argument of 6’h2A (42). Send the ADC_TACQ_Rd command and read back this commanded data argument.
4. Send the ADC_TACQ_Wr command with a data argument of 6’h3F (63). Send the ADC_TACQ_Rd command and read back this commanded data argument.
5. Send the ADC_TACQ_Wr command with a data argument of 6’h0 (0). Send the ADC_TACQ_Rd command and read back this commanded data argument.
6. Send the GARC_Reset command to ensure all registers are at the proper initial value.

6.34 GAFE ASICs Mode Register Test

This section tests the proper functioning of the GAFE mode register. Repeat for all GAFE ASICs present using valid GAFE addressing.

1. Send the GAFE_Mode_Write command to the GAFE being tested with a data field of 16’h00 (0).
2. Send the GAFE_Mode_Read command. The data field in the return data stream should be 16’h00 (0).
3. Send the GAFE_Mode_Write command to the GAFE being tested with a data field of 16’hFF (255).

4. Send the GAFE_Mode_Read command. The data field in the return data stream should be 16’hFF (255).
5. Send the GAFE_Mode_Write command to the GAFE being tested with a data field of 16’h30 (48).
6. Send the GAFE_Mode_Read command. The data field in the return data stream should be 16’h30 (48).

6.35 GAFE ASIC VETO DAC Register Test

This section tests the proper functioning of the GAFE DAC #1 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC1_Write command to the GAFE being tested with a data field of 16’h00 (0).
2. Send the GAFE_DAC1_Read command. The data field in the return data stream should be 16’h00 (0).
3. Send the GAFE_DAC1_Write command to the GAFE being tested with a data field of 16’h3F (63).
4. Send the GAFE_DAC1_Read command. The data field in the return data stream should be 16’h3F (63).
5. Send the GAFE_DAC1_Write command to the GAFE being tested with a data field of 16’h39 (57).
6. Send the GAFE_DAC1_Read command. The data field in the return data stream should be 16’h39 (57).

6.36 GAFE ASIC VETO VERNIER Register Test

This section tests the proper functioning of the GAFE DAC #2 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC2_Write command to the GAFE being tested with a data field of 16’h00 (0).
2. Send the GAFE_DAC2_Read command. The data field in the return data stream should be 16’h00 (0).
3. Send the GAFE_DAC2_Write command to the GAFE being tested with a data field of 16’h3F (63).
4. Send the GAFE_DAC2_Read command. The data field in the return data stream should be 16’h3F (63).
5. Send the GAFE_DAC2_Write command to the GAFE being tested with a data field of 16’h26 (38).
6. Send the GAFE_DAC2_Read command. The data field in the return data stream should be 16’h26 (38).

6.37 GAFE ASIC HLD Register Test

This section tests the proper functioning of the GAFE DAC #3 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC3_Write command to the GAFE being tested with a data field of 16’h00 (0).

2. Send the GAFE_DAC3_Read command. The data field in the return data stream should be 16’h00 (0).
3. Send the GAFE_DAC3_Write command to the GAFE being tested with a data field of 16’h3F (63).
4. Send the GAFE_DAC3_Read command. The data field in the return data stream should be 16’h3F (63).
5. Send the GAFE_DAC3_Write command to the GAFE being tested with a data field of 16’h37 (55).
6. Send the GAFE_DAC3_Read command. The data field in the return data stream should be 16’h37 (55).

6.38 GAFE ASIC BIAS Register Test

This section tests the proper functioning of the GAFE DAC #4 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC4_Write command to the GAFE being tested with a data field of 16’h00 (0).
2. Send the GAFE_DAC4_Read command. The data field in the return data stream should be 16’h00 (0).
3. Send the GAFE_DAC4_Write command to the GAFE being tested with a data field of 16’h3F (63).
4. Send the GAFE_DAC4_Read command. The data field in the return data stream should be 16’h3F (63).
5. Send the GAFE_DAC4_Write command to the GAFE being tested with a data field of 16’h20 (32).
6. Send the GAFE_DAC4_Read command. The data field in the return data stream should be 16’h20 (32).

6.39 GAFE ASIC TCI Register Test

This section tests the proper functioning of the GAFE DAC #5 register. Repeat for all GAFE ASICs present, using valid GAFE addressing.

1. Send the GAFE_DAC5_Write command to the GAFE being tested with a data field of 16’h00 (0).
2. Send the GAFE_DAC5_Read command. The data field in the return data stream should be 16’h00 (0).
3. Send the GAFE_DAC5_Write command to the GAFE being tested with a data field of 16’h3F (63).
4. Send the GAFE_DAC5_Read command. The data field in the return data stream should be 16’h3F (63).
5. Send the GAFE_DAC5_Write command to the GAFE being tested with a data field of 16’h00 (0).
6. Send the GAFE_DAC5_Read command. The data field in the return data stream should be 16’h00 (0).

At the successful conclusion of this section, the GARC commandable register bits have been demonstrated to be functional with all bits toggling as commanded.

6.40 Test of the GARC Parity Error Detection and Error Generation

This section tests the proper functioning of the GARC return data parity select bit and the GARC's ability to detect command and data parity errors.

This test may be automated using the LabView GSE via the **GARC Parity Test.txt** script.

1. Send the GARC_Reset command.
2. Send the GARC_Status command. The data field in the return data stream should be decimal 24.
3. Send the GARC_Mode command with a data argument of decimal 769, a command to return event data with even parity.
4. Send the GARC_Status command. The data field in the return data stream should show an AEM parity error.
5. Send the GARC_Mode command with a data argument of decimal 768, a command back to odd parity, the nominal mode.
6. Send the GARC_Status command. The data field in the return data stream should be decimal 24. The AEM should indicate nominal parity (no errors).
7. Send the GARC_Reset command to reinitialize all GARC registers. Ready to test Command and Data parity errors.
8. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b0000.
9. Send the GARC_Version command.
10. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b0000.
11. Send the Command_Register command and verify that the return data is 16'h0000.
12. Send the even command parity (i.e., error) command: 34'h2404C0001.
13. Send the GARC_Diagnostic command and verify that bits 15:12 in the return data word are 4'b1101.
14. Send the Command_Register command and verify that the return data is 16'h404E.
15. At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

6.41 Test of the GARC Diagnostic Status Register

This test may be automated using the LabView GSE via the **GARC_Diagnostic_Status_Reg_Test.tst** script.

1. Send the GARC_Reset command to initialize the GARC registers.
2. Send the GARC_Mode_Rd command. The data field in the return data stream should be decimal 768.
3. Send the GARC_Status command. The data field in the return data stream should be decimal 24.
4. Send the GARC_Cmd_Reg command. The data field in the return data stream should be decimal 0.
5. Send the GARC_Diagnostic command. The most significant four bits in the return data stream are the Diagnostic bits. The remaining 12 bits are state machine loop and command counters and the total should not be zero. The data field in the return data stream should be decimal 771.

6.42 Command Counter Test

This test may be automated using the LabView GSE via the **GARC_Command_Counter_Test.txt** script.

1. Send the GARC_Reset command to initialize the GARC registers.
2. Send the GARC_Diagnostic command. The value in bits 7:0 of the returned data word should have the value 0.
3. Send the GARC_Version command ten times.
4. Send the GARC_Diagnostic command. The value of the returned data word should have the value 2827.

6.43 Test of the Look-At-Me Circuitry

This section tests the proper functioning of the GARC Look-At-Me circuitry. The GARC has the capability to receive commands from either the primary side or secondary side. The selection of which set of receivers to listen to is controlled by the Look-At-Me circuitry. This circuitry toggles the status of the receiving side based upon receipt of a special command pattern (e.g., **34'h24153D721**). This the equivalent of sending a GARC configuration command to addr 1, function 4, with a data pattern of 60304.

Another way to look at the bit pattern (in a format like the ICD would present it) would be:

1001	==	configuration command
0	==	GARC
00001	==	address 1
0	==	write
1	==	a command with data
0100	==	function 4
1	==	command parity
16'h	==	EB90 == data field (60304)
0	==	data parity

This test may be automated using the LabView GSE via the GARC Look At Me Test.txt script.

Turn on the system and send a Look-at-me command. Send the GARC_Status command. The data field in the return data stream should be decimal 24. Bit 0 (LSB) of the Status register represents the Look-At-Me status (0 = A, 1 = B).. Set the GSE to the Secondary side, Send the GARC_Status command. Observe that there is no response from the GARC.
Send the Look_At_Me command to the GARC from the secondary side interface. Send the GARC_Status command. The data field in the return data should be decimal 25 (with the LSB = 1, indicating the Look-At-Me status is B side). Set the GSE for the Primary side and send the GARC_Status command. Observe that there is no response from the GARC. Send the Look_At_Me command to the GARC from the primary side interface. Send the GARC_Status command. The data field in the return data should be decimal 24 (with the LSB = 0, indicating the Look-At-Me status is A side).

This concludes the test of the GARC Look-At-Me circuitry. At the successful conclusion of this section, please document this fact by making an entry in the copy of the Comprehensive Performance Test Record associated with this procedure.

7.0 Test of the GARC LVDS Circuitry Driver Currents

This section tests the proper functioning of the GARC LVDS Driver circuitry. It is required that all LVDS drivers be terminated at the receiver with a 100 ohm resistor. The ACD-to-AEM nominal drive current is 3.5 mA across the 100 ohms for a voltage differential of 350 mV.

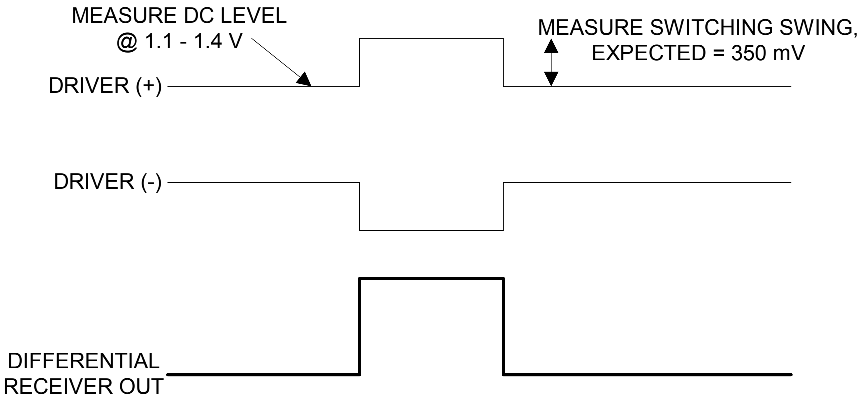
There are 6 LVDS receiver input pairs to the GARC:

a) ACD_CLK_A, ACD_CLK_B
b) ACD_CMD_A, ACD_CMD_B
c) ACD_RESET_A, ACD_RESET_B

There are 40 LVDS driver output pairs from the GARC:

d) ACD_NSDATA_A, ACD_NSDATA_B
e) ACD_NVETO_nnA, ACD_NVETO_nnB (where nn is 00 – 17)
f) ACD_CNO_A, ACD_CNO_B

The "A" side interface will be examined first. Verify that an AEM (or AEM simulator) is connected to the GARC "A" side interface. Using a digitizing oscilloscope, measure the DC baseline and switching differential voltage on each of the driver/receiver pairs. Two scope probes may be used, one on each side of the differential receiver inputs. The DC level and switching voltages may be measured as diagrammed below. The measurement will be made with the Drivers enabled and with them disabled.



Record the information for each pin on the Primary side interface in the table below. Please document the data from the following steps in the copy of the Test Record associated with this procedure.

Obtain a scope picture of the nominal pulse Amplitude Swings

GARC Signal	Signal Location	- DC Level (V) Enabled	+ DC Level (V) Enabled	- DC Level (V) Disabled	+ DC Level (V) Disabled
ACD_NVETO_12B	R1				
ACD_NVETO_12A	R2				
ACD_NVETO_11B	R3				
ACD_NVETO_11A	R4				
ACD_NVETO_10B	R5				
ACD_NVETO_10A	R6				
ACD_NVETO_09B	R7				
ACD_NVETO_09A	R8				
ACD_NVETO_08B	R9				
ACD_NVETO_08A	R10				

ACD_NVETO_07B	R11				
ACD_NVETO_07A	R12				
ACD_NVETO_06B	R13				
ACD_NVETO_06A	R14				
ACD_NVETO_05B	R15				
ACD_NVETO_05A	R16				
ACD_NVETO_04B	R17				
ACD_NVETO_04A	R18				
ACD_NVETO_03B	R19				
ACD_NVETO_03A	R20				
ACD_NVETO_02B	R21				
ACD_NVETO_02A	R22				
ACD_NVETO_01B	R23				
ACD_NVETO_01A	R24				
ACD_NVETO_00B	R25				
ACD_NVETO_00A	R26				
ACD_CNO_B	R27				
ACD_CNO_A	R28				
ACD_NVETO_13B	R31				
ACD_NVETO_13A	R32				
ACD_NVETO_14B	R33				
ACD_NVETO_14A	R34				
ACD_NVETO_15B	R35				
ACD_NVETO_15A	R36				
ACD_NVETO_16B	R37				
ACD_NVETO_16A	R38				
ACD_NVETO_17B	R39				
ACD_NVETO_17A	R40				
ACD_NSDATA_B	R29				
ACD_NSDATA_A	R30				

Set the pulse generator for input to J3-3 for 3.3V and 1uSec. J3-6 is Gnd.

Measure the GAFE to GARC LVDS signals if the GAFE Simulator is being used.

GARC Signal	Test Board Pin	DC Level (V)	Switch Swing (V)
IRTN_00	J4-33		
DISC_00	J4-34		
CHID_00	J4-35		
IRTN_01	J4-36		
DISC_01	J4-37		
CHID_01	J4-38		
IRTN_02	J4-39		
DISC_02	J4-40		
CHID_02	J4-41		
IRTN_03	J4-42		
DISC_03	J4-43		
CHID_03	J4-44		
IRTN_04	J4-45		
DISC_04	J4-46		
CHID_04	J4-47		
IRTN_05	J4-48		
DISC_05	J4-49		
CHID_05	J4-50		

IRTN_06	J4-51		
DISC_06	J4-52		
CHID_06	J4-53		
IRTN_07	J4-54		
DISC_07	J4-55		
CHID_07	J4-56		
IRTN_08	J4-57		
DISC_08	J4-58		
CHID_08	J4-59		
IRTN_09	J4-60		
DISC_09	J4-61		
CHID_09	J4-62		
IRTN_10	J4-63		
DISC_10	J4-64		
CHID_10	J4-65		
IRTN_11	J4-66		
DISC_11	J4-67		
CHID_11	J4-68		
IRTN_12	J4-69		
DISC_12	J4-70		
CHID_12	J4-71		
IRTN_13	J4-72		
DISC_13	J4-73		
CHID_13	J4-74		
IRTN_14	J4-75		
DISC_14	J4-76		
CHID_14	J4-77		
IRTN_15	J4-78		
DISC_15	J4-79		
CHID_15	J4-80		
IRTN_16	J4-81		
DISC_16	J4-82		
CHID_16	J4-83		
IRTN_17	J4-84		
DISC_17	J4-85		
CHID_17	J4-86		

This completes the testing of the GARC LVDS driver circuitry.

8.0 Power Supply Rail Tests – 3.6V to 3.0V

The function of this test is to verify operation of the GARC circuitry over the full Power Supply Rail limits. This will be verified by performing the register test. This may be done via automated script.

1) Set power supply to +3.6V high limit.
2) Record the +3.6V Current

+3.6V current - _______________________

3) Run the register test from section 6
4) Verify that all parameters are essentially unchanged

Register Test Verified - ___________________

5) Set power supply to +3.0V high limit.
6) Record the +3.0V Current

+3.0V current - _______________________

7) Run the register test from section 6
8) Verify that all parameters are essentially unchanged

Register Test Verified - ___________________

Appendix 1: GARC Documentation

A more complete set of documentation (such as the Verilog, EDIF, layout, and wirebonding diagram) is available on the LHEA ACD electronics web page at:

http://lhea-glast.gsfc.nasa.gov/acd/electronics/

Appendix 2: GARC Configuration Command Format

The GARC logic core responds only to properly structured commands. There are two possible command types – Trigger Commands and Configuration Commands. Trigger commands initiate an Event Data cycle, causing a GAFE Hold, an analog-to-digital conversion, and the return of an event data packet. Configuration commands are used to either read or write GARC or GAFE registers.

The format for Trigger Commands is detailed in the table below.

Bit(s)	Bit Description	Value
3	Start Bit	1
2:1	Trigger Type Bits	10 for ZS Enabled Trigger 01 for Send All PHA Trigger
0	Parity Bit	Odd parity bit over previous two bits

Therefore, a 1100 is a ZS Enabled Trigger command and a 1010 is a Send All PHA Trigger command.

This format for GARC Configuration Commands is detailed in the table below.

Field	# bits	Function
Start	1	1 for start
CMD Type	2	00 for command
CMD Type Parity	1	Odd Parity over previous 2 bits (without Start bit)
GAFE/GARC Select	1	0 for GARC 1 for GAFE
GAFE/GARC Address	5	GAFE: Select which GAFE, 0x1F for all GAFE GARC: Select which function block
Read/Write	1	0 for write, 1 for read
Data/Dataless	1	0 for dataless, 1 for data, always 1 for ACD
register/function number	4	Which register/function in the function block
CMD Parity	1	Odd parity bit over previous 15 bits
Data	16	Data Field
Data Parity	1	Odd parity bit over previous 16 bits

Appendix 3: GARC Event Data Format:

The GARC core returns an event data packet when a valid trigger command is received. The event data format is detailed in the table below.

Field	# bits	Function
Start Bit	1	1 for start
Hit Map Bits	18	Bits 17-0 for channels 0-17, bit set if hit in channel
Zero Suppression Bits	18	Bits 17-0 for channels 0-17, bit set if PHA above threshold
CMD/Data ERROR	1	Error in command parity detected
Header Parity	1	Odd parity bit over previous 37 bits
PHA Words (quantity 0-18) Order: Channel 0 to channel 17	15	Bit 14: 1 if another PHA word follows this one, 0 if this is last one Bit 13: 1 for high range, 0 for low range Bits 12-1: the PHA value, 0 to 4095 Bit 0: Odd parity over last 14 bits

Appendix 4: GARC Configuration Data Readback Format:

The GARC core returns an register configuration data packet when a valid configuration readback command is received. The configuration readback data format is detailed in the table below.

Field	# bits	Function
Start	1	1 for start
GAFE/GARC Select	1	Copy of write command field (0 for GARC, 1 for GAFE)
GAFE/GARC Address	5	Copy of write command field (Select which GAFE)
Read/Write	1	1 for read
Data/Dataless	1	always 1
Register/function number	4	Copy of write command field (which register/function in the function block)
CMD Parity	1	Odd parity bit over previous 12 bits
Data	16	Data, MSB first
CMD/DATA ERROR	1	Error in parity detected
Parity	1	Odd parity bit over previous 17 bits

Appendix 5: Test Results Record

All measurements and test results will be recorded in this Test Results Record. A copy of the entire this entire document is not required each time the test is performed. The appendix will serve as the official record each time this test is performed.

2.0 QA Signoff

Quality Assurance approval to proceed with the test - ______________________
(NOTE: If QA gives verbal approval to proceed but is not in attendance for the test, record the date and the time of approval.)

BIAS Resistor Verified on the GARC Test Board - __________________

2.1 GARC Identification

The test conductor for this test is: ___________________________

Date of Test: ___________________________

The identification/Serial Number listed on the GARC ASIC is: ___________________________

The serial number of the AEM Simulator board is: ___________________________

The serial number of the GARC Test board is: ___________________________

The serial number of the GAFE Simulator board is: ___________________________

2.2 Test Equipment Utilized

Instrument Type	Manufacturer & Model Number	NASA ID Number	Calibration Due Date
Power Supply +3.3V			
Power Supply +5.0V			
Multimeter 1			
Multimeter 2			
Pulse Generator			
Oscilloscope			

4.1 Measurement of the GARC Bias Resistors and Voltages

+3.3V Current - ________________________(_______± 5mA)

GARC Bias Signal	GARC Pin	Resistor Test Point	Expected Voltage	Measured Voltage
HLD_WOR_BIAS	104	R19	1.64V	
BIAS_RCVR	156	R16	1.10V	
BIAS_DRV_H	160	R15	1.53V	
BIAS_DRV_L	169	R13	1.75V	
LVDS_PRESET_ADJ	184	R20	1.52V	

4.2 GARC Registers Initial Reset Test

Turn On reset verified - ______________________

Board Serial Number verified - __________________

Reset Command verified - ____________________

5.1 FREE Power Measurement at the Nominal Power Supply Voltage

GARC Mode	GARC_Mode_Wr Data Argument	+3.3V Current Measured (mA)	+3.3V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		185 ± 10mA
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		135 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		135 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		80 ± 10mA

5.2 FREE Power Measurement at the Minimum Power Supply Voltage

GARC Mode	GARC_Mode_Wr Data Argument	+3.0V Current Measured (mA)	+3.0V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		160 ± 10mA
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		115 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		115 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		65 ± 10mA

5.3 FREE Power Measurement at the Maximum Power Supply Voltage

GARC Mode	GARC_Mode_Wr Data Argument	+3.6V Current Measured (mA)	+3.6V Current Expected (mA)
LVDS "A" Drivers Enabled LVDS "B" Drivers Enabled	768		215 ± 10mA
LVDS "A" Drivers Enabled LVDS "B" Drivers Disabled	256		155 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Enabled	512		155 ± 10mA
LVDS "A" Drivers Disabled LVDS "B" Drivers Disabled	0		95 ± 10mA

6.0 GARC Register Read/Write Tests

This test may be automated using the LabView GSE via the Register Test VI.

Test Completed - __________________

6.40 Test of the GARC Parity Test

This test may be automated using the LabView GSE via the **GARC Parity Test.txt** script

Test Completed - __________________

6.41 Test of the GARC Diagnostic Register

This test may be automated using the LabView GSE via the **GARC Diagnostic Status Reg Test.txt** script

Test Completed - __________________

6.42 Test of the GARC Command Counters

This test may be automated using the LabView GSE via the **GARC Command Counter Test.txt** script

Test Completed - __________________

6.43 Test of the Look-At-Me Circuitry

This test may be automated using the LabView GSE via the **GARC Look At Me Test.txt** script.

Test Completed - __________________

7.0 Test of the GARC LVDS Circuitry Driver Currents

Obtain a scope picture of the nominal pulse Amplitude Swings

GARC Signal	Signal Location	- DC Level (V) Enabled	+ DC Level (V) Disabled	- DC Level (V) Enabled	+ DC Level (V) Disabled
ACD_NVETO_12B	R1				
ACD_NVETO_12A	R2				
ACD_NVETO_11B	R3				
ACD_NVETO_11A	R4				
ACD_NVETO_10B	R5				
ACD_NVETO_10A	R6				
ACD_NVETO_09B	R7				
ACD_NVETO_09A	R8				
ACD_NVETO_08B	R9				
ACD_NVETO_08A	R10				
ACD_NVETO_07B	R11				
ACD_NVETO_07A	R12				
ACD_NVETO_06B	R13				
ACD_NVETO_06A	R14				
ACD_NVETO_05B	R15				
ACD_NVETO_05A	R16				
ACD_NVETO_04B	R17				
ACD_NVETO_04A	R18				
ACD_NVETO_03B	R19				
ACD_NVETO_03A	R20				
ACD_NVETO_02B	R21				
ACD_NVETO_02A	R22				
ACD_NVETO_01B	R23				
ACD_NVETO_01A	R24				
ACD_NVETO_00B	R25				
ACD_NVETO_00A	R26				
ACD_CNO_B	R27				
ACD_CNO_A	R28				
ACD_NVETO_13B	R31				
ACD_NVETO_13A	R32				
ACD_NVETO_14B	R33				
ACD_NVETO_14A	R34				
ACD_NVETO_15B	R35				
ACD_NVETO_15A	R36				
ACD_NVETO_16B	R37				
ACD_NVETO_16A	R38				
ACD_NVETO_17B	R39				
ACD_NVETO_17A	R40				
ACD_NSDATA_B	R29				
ACD_NSDATA_A	R30				

Verify the signal at J3-3 is +3.3V and 1uSec wide. Gnd is on J3-6.

Verified - __________________

Measure the GAFE to GARC LVDS signals.

GARC Signal	Test Board Pin	DC Level (V)	Switch Swing (V)
IRTN_00	J4-33		
DISC_00	J4-34		
CHID_00	J4-35		
IRTN_01	J4-36		
DISC_01	J4-37		
CHID_01	J4-38		
IRTN_02	J4-39		
DISC_02	J4-40		
CHID_02	J4-41		
IRTN_03	J4-42		
DISC_03	J4-43		
CHID_03	J4-44		
IRTN_04	J4-45		
DISC_04	J4-46		
CHID_04	J4-47		
IRTN_05	J4-48		
DISC_05	J4-49		
CHID_05	J4-50		
IRTN_06	J4-51		
DISC_06	J4-52		
CHID_06	J4-53		
IRTN_07	J4-54		
DISC_07	J4-55		
CHID_07	J4-56		
IRTN_08	J4-57		
DISC_08	J4-58		
CHID_08	J4-59		
IRTN_09	J4-60		
DISC_09	J4-61		
CHID_09	J4-62		
IRTN_10	J4-63		
DISC_10	J4-64		
CHID_10	J4-65		
IRTN_11	J4-66		
DISC_11	J4-67		
CHID_11	J4-68		
IRTN_12	J4-69		
DISC_12	J4-70		
CHID_12	J4-71		
IRTN_13	J4-72		
DISC_13	J4-73		
CHID_13	J4-74		
IRTN_14	J4-75		
DISC_14	J4-76		
CHID_14	J4-77		
IRTN_15	J4-78		
DISC_15	J4-79		
CHID_15	J4-80		
IRTN_16	J4-81		
DISC_16	J4-82		
CHID_16	J4-83		
IRTN_17	J4-84		
DISC_17	J4-85		

CHID_17	J4-86		

8.0 Power Rail Tests

2) +3.6V Current - ____________________________

Register Test Verified - _____________________

6) +3.0V Current - ____________________________

Register Test Verified - _____________________

END OF GARC SHORT FUNCTIONAL TEST PROCEDURE